BLUETOOTH DOC	Date / Year-Month-Day 2008-06-26	Approved	Revision V10r00	Document No HDP_SPEC
Prepared Medical Devices WG	e-mail address med-feedback@bluetooth.org			N.B.

HEALTH DEVICE PROFILE

Abstract

This profile together with the Multi-Channel Adaptation Protocol (MCAP) enables Healthcare and Fitness device usage models.

Revision History

Revision	Date	Comments
D05r09	24 January 2007	Revision 0.5 approved by BARB
D09r05	29 September 2007	Revision 0.9 approved by BARB
D10r14	18 June 2008	Revision 1.0 approved by BARB
D10r15	25 June 2008	Prepare for publication
V10r00	26 June 2008	Adopted by the Bluetooth SIG Board of Directors

Contributors

Company	Name
A & D Medical	Jerry Wang
Bluegiga	Hermanni Suominen
Broadcom	Victor Zhodzishsky
connectBlue	Mats Andersson
Cybercom	Pär Sandell Ulf Karlsson
Ezurio	Nick Hunn
IBM	Michael Nidd
Intel Corporation	Robert D. Hughes (editor) Doug Bogia
MindTree	Dennis Mathews
Motorola	Fabio Grigorjev
Nonin	Jayant Parthasarathy
Philips	Lars Schmitt
Radio, Digital Technique	Brad Tipler
Socket Mobile	Len Ott
Stollmann	Karsten Aalders Juergen Wichmann
Symbian	Tim Howes
Welch Allyn	Jim DelloStritto

Disclaimer and Copyright Notice

The copyright in this specification is owned by the Promoter Members of *Bluetooth*® Special Interest Group (SIG), Inc. (“*Bluetooth* SIG”). Use of these specifications and any related intellectual property (collectively, the “Specification”), is governed by the Promoters Membership Agreement among the Promoter Members and *Bluetooth* SIG (the “Promoters Agreement”), certain membership agreements between *Bluetooth* SIG and its Adopter and Associate Members (the “Membership Agreements”) and the *Bluetooth* Specification Early Adopters Agreements (1.2 Early Adopters Agreements) among Early Adopter members of the unincorporated *Bluetooth* SIG and the Promoter Members (the “Early Adopters Agreement”). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to *Bluetooth* SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of *Bluetooth* SIG or a party to an Early Adopters Agreement (each such person or party, a “Member”), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by applicable law to *Bluetooth* SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the *Bluetooth* technology ("*Bluetooth* products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of *Bluetooth* products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their *Bluetooth* Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their *Bluetooth* products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses**. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.**

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST *BLUETOOTH* SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

Document Terminology

The Bluetooth SIG has adopted Section 13.1 of the IEEE Standards Style Manual [6], which dictates use of the words ``shall'', ``should'', ``may'', and ``can'' in the development of documentation, as follows:

- The word **shall** is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (shall equals is required to).
- The word **should** is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (should equals is recommended that).
- The word **may** is used to indicate a course of action permissible within the limits of the standard (may equals is permitted).
- The word **can** is used for statements of possibility and capability, whether material, physical, or causal (can equals is able to).
- The use of the word **must** is deprecated and shall not be used when stating mandatory requirements; must is used only to describe unavoidable situations.
- The use of the word **will** is deprecated and shall not be used when stating mandatory requirements; will is only used in statements of fact.

Contents

1 Introduction

1.1 Scope

The Health Device Profile (HDP) is an application profile that defines the requirements for qualified Bluetooth Healthcare and Fitness (referred to as 'health') device implementations. This profile is used for connecting application data *Source* devices such as blood pressure monitors, weight scales, glucose meters, thermometers, and pulse oximeters to application data *Sink* devices such as mobile phones, laptops, desktop computers, and health appliances without the need for cables. This profile makes use of the Multi-Channel Adaptation Protocol (MCAP) [2] and new L2CAP features such as Enhanced Retransmission Mode, Streaming Mode and optional FCS [1] to define the interoperability requirements.

This revision of HDP together with MCAP provides the following:

- Provisions for a standardized, structured approach for using a Control Channel to connect and coordinate connection of necessary Data Channels. This allows retention of state references between reconnections and provides for the structured management of a group of related channels between pairs of devices
- Optional synchronization of timers with microsecond accuracy in participating devices, allowing time stamps to be directly compared
- Defines an efficient method of reconnection for power conservation

HDP is specialized for health applications and thus has the following advantages over other more generic profiles:

- Defines the interoperability requirements for the applications within the HDP application
- Provides strong application level interoperability by operating with the ISO/IEEE 11073-20601 Personal Health Data Exchange Protocol [7], which defines a transport-agnostic Data Exchange Protocol and representation of device application data based on international standards.
- Provides strong application level interoperability by operating with Device Data Specializations which are compatible with the Data Exchange Protocol [7]. The Device Data Specializations define the format of device application data based on international standards and can be added through updates to the Bluetooth Assigned Numbers [3] without the need to update the HDP specification.
- Provisions for a standardized method by which the device-type and supported application data-types of a device can be determined wirelessly, using the Bluetooth Service Discovery Protocol (SDP).
- Connection-oriented to ensure more reliable behavior when a *Source* moves out of range or disconnects (either inadvertently or intentionally), allowing the device to recognize the condition and take appropriate actions.
- Permits multiple simultaneous Data Channels.

- Allows per-channel Logical Link Control and Adaptation Protocol (L2CAP) [1] configuration using Enhanced Retransmission Mode and Streaming Mode which allows Data Channels to be configured separately from each other and from the Control Channel. This allows for flexibility to configure channel reliability as required.
- Relatively inexpensive to implement since it is based on MCAP which has a small list of relatively simple control commands and low code-space requirements.

The HDP is not intended to support:

- Transmission of health-related audio from devices such Stethoscopes; it is expected that audio applications are best enabled through the use of profiles specializing in this type of application.

1.2 Conformance

If conformance to this profile is claimed, all capabilities indicated as mandatory for this profile shall be supported in the specified manner (process mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory, optional and conditional capabilities, for which support is indicated, are subject to verification as part of the Bluetooth Qualification Program.

1.3 Profile and Protocol Dependencies

Devices that claim compliance to this profile shall not have dependency on any profiles or protocols other than described in this section. A profile/protocol is dependent upon another if it re-uses parts of that profile/protocol, by explicitly referencing it. In the case of HDP, it is dependent upon MCAP and Generic Access Profile and has conditional dependence on the Device ID Profile (refer to Section 5.3).

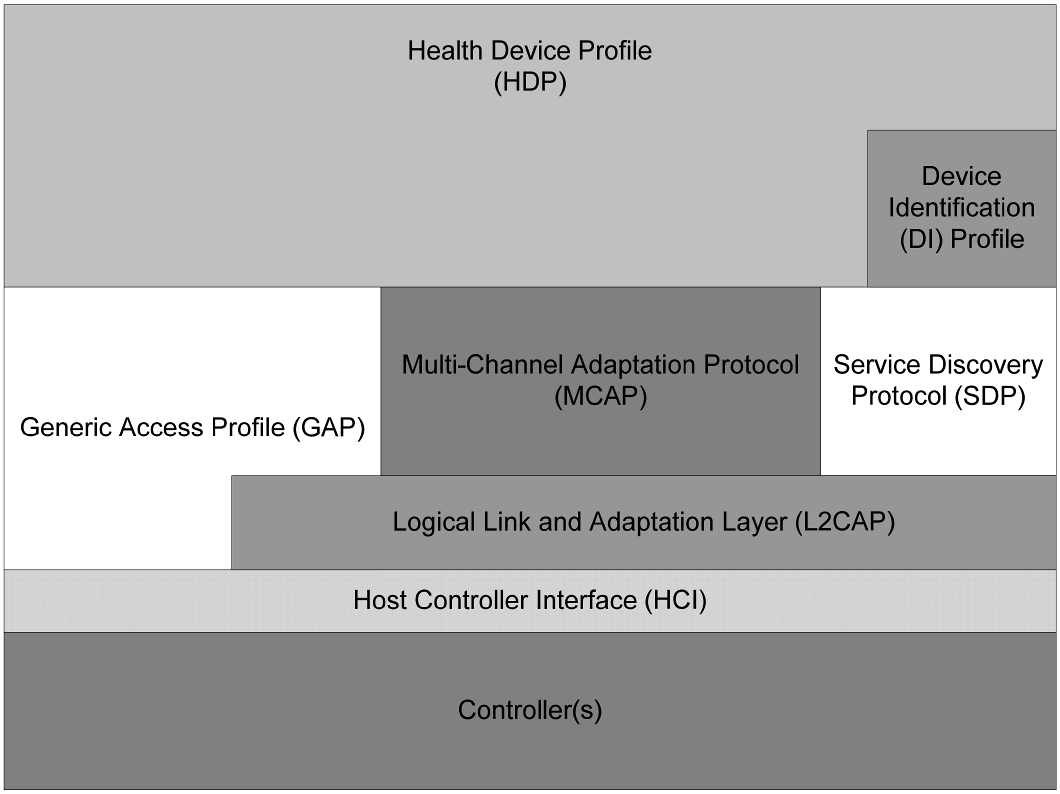
Health Device Profile
(HDP)
Device
Identification
(DI) Profile
Generic Access Profile (GAP)
Multi-Channel Adaptation Protocol
(MCAP)
Service Discovery
Protocol (SDP)
Logical Link and Adaptation Layer (L2CAP)
Host Controller Interface (HCI)
Controller(s)

Figure 1.1: Profile and Protocol Dependency Model

1.4 Symbols and Conventions

1.4.1 Requirement Status Symbols

In this document the following symbols are used:

M = Mandatory to support (used for capabilities that shall be implemented in the profile)

O = Optional to support (used for capabilities that may be implemented in the profile)

C = Conditional to support (used for capabilities that shall be implemented in case a certain other capability is supported)

X = Excluded (used for capabilities that may be supported by the unit but shall never be used in the profile)

N/A = Not Applicable (in the given context it is impossible to use this capability)

Some excluded capabilities (identified as X) are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

1.5 Related Specifications

1.5.1 Core Specification

"Core Specification" refers to the Bluetooth Core Specification 2.0 + EDR or 2.1 + EDR with Volume 3, Part A of Core Specification Addendum 1 [1] or later versions of the Bluetooth Core Specification adopted by the Bluetooth® SIG.

1.5.2 Multi-Channel Adaptation Protocol (MCAP)

MCAP refers to the Multi-Channel Adaptation Protocol specification [2].

1.5.3 Data Exchange Specifications

"Data Exchange Specifications" refers to the Data Exchange Protocol in conjunction with the Device Data Specializations. These facilitate interoperability between sources and recipients of device application data, minimizing the need for proprietary drivers on either side to format and interpret the application data.

"Data Exchange Protocol" refers to the following specification, which (in turn) references other required specifications:

- IEEE Std 11073-20601 ™ -2008 Health informatics - Personal health device communication - Application profile - Optimized exchange protocol

Relative to the Data Exchange Protocol specification defined above, applicable "Device Data Specializations" are listed in the Bluetooth Assigned Numbers [3]. Refer to Section 4 for Data Exchange Specification requirements.

2 Profile Overview

This section is informative to assist the reader in understanding the remainder of this document. For a more detailed description of HDP including usage examples, please refer to the HDP Implementation Guidance Whitepaper [12].

2.1 Protocol Stack

Figure 2.1 shows the protocols and entities used in this profile.

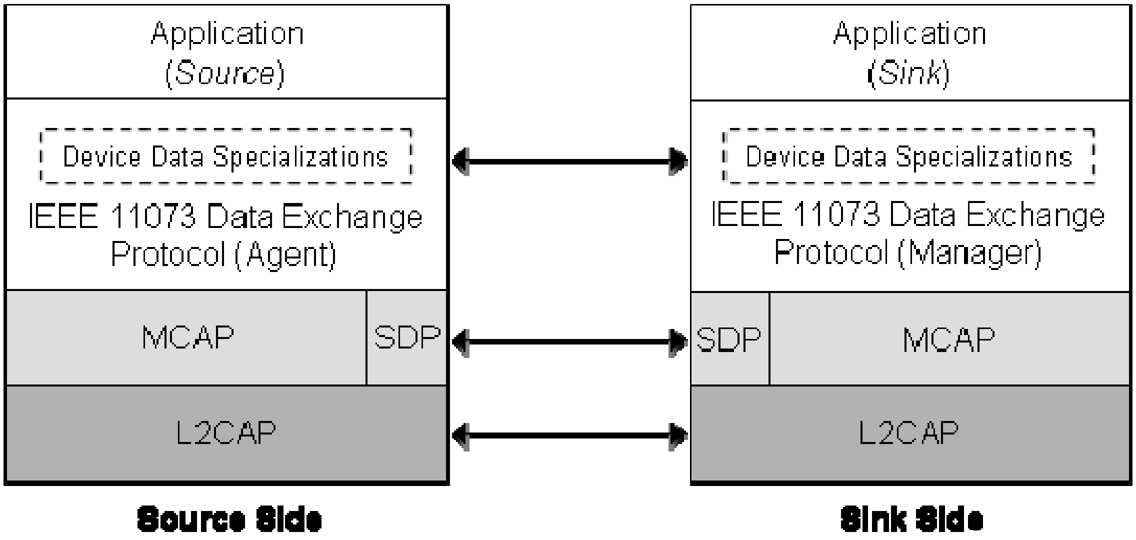
Application
(Source)
Device Data Specializations
IEEE 11073 Data Exchange Protocol (Agent)
MCAP
SDP
L2CAP
Source Side
Application
(Sink)
Device Data Specializations
IEEE 11073 Data Exchange Protocol (Manager)
SDP
MCAP
L2CAP
Sink Side

Figure 2.1: Protocol Model

HDP depends upon MCAP as its transport protocol and references Data Exchange Specifications to define the data-layer.

2.2 Configurations and Roles

The following terms are used to define the roles and configurations of the profile. These terms are defined completely in MCAP [2] and are listed below for reference.

Source / Sink
A *Source* is a transmitter of application data that is to be transferred to a *Sink*.

A *Sink* is a receiver of application data delivered from a *Source*.

Initiator / Acceptor
The device that initiates the Control Channel connection is referred to as the Initiator of the connection.

The device that is addressed in a Control Channel connection request is the Acceptor of the connection.

Control Channel / Data Channel

The first L2CAP channel established between two instances of MCAP is the “Control Channel.” This channel facilitates the creation of “Data Channels,” over which application data defined in the Data Exchange Specifications can be exchanged. The L2CAP channel configuration requirements for the Control and Data Channels are defined as part of this profile.

MCAP Data End Point (MDEP)
A MCAP Data End Point (MDEP) represents the logical function of a device and includes such information as MDEP ID, MDEP Role (*Source*/*Sink*), Data Type (Device Data Specialization) and Description.

MCAP Data Link (MDL)
An MCAP Data Link (MDL) identifies a pair of MDEPs, and is explicitly created by one of the two participating devices as a result of a request on the Control Channel.

MCAP Communications Link (MCL)
An MCAP Communications Link (MCL) refers to the collection of L2CAP connections between two instances of MCAP and is comprised of a Control Channel and zero[1] or more Data Channels.

2.2.1 Example Applications

The figures that follow show examples for which the HDP is applicable. Note that each link in these examples contains one Control Channel and one or more Data Channels. Throughout this section, the following notation is used:

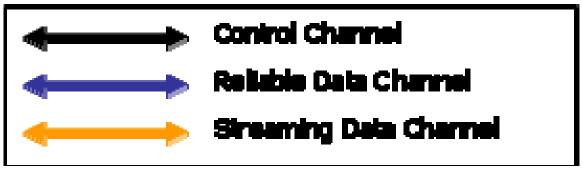

Figure 2.2: Figure Notation

The Data Channel payload may be known *a priori* (as with periodic or batch transmissions) or the connection may remain open for an indefinite period of time (as with streaming data). To suit a variety of needs, Data Channels can be configured as Streaming Data Channels or Reliable Data Channels; however, at least one Reliable Data Channel is required. Examples of data transmitted on Streaming Data Channels may include streaming application data from pulse oximeters, electrocardiograms (ECGs) and electroencephalograms (EEGs). Examples of data transmitted on Reliable Data Channels may include episodic or periodic application data from blood pressure monitors, thermometers, heart rate monitors, glucose meters and weight scales. Either or both may be utilized by devices that provide an aggregation, routing or store and forward function to another *Source* device.

[1] If Standard Op Codes are supported, then zero is typically a transitional state after a Control Channel is established and before a Data Channel is established. The Clock Synchronization Protocol can function normally in the absence of any Data Channels.

Figure 2.3 shows an example scenario in which application data from a *Source* device with a Reliable Data Channel is transmitted to a *Sink*. Examples of these types of *Source devices* may include weight scales, blood pressure meters, thermometers and glucose meters. Examples of *Sinks* may include PCs, mobile phones, PDAs and Telehealth stations.

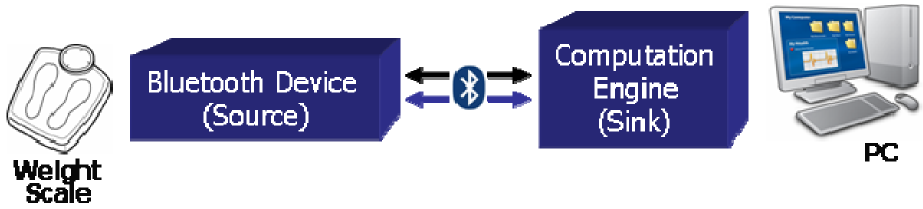

Figure 2.3: Example HDP Use - Reliable Data Channel Data

Figure 2.4 shows an example scenario in which application data from a *Source* device with a Streaming Data Channel is transmitted to a *Sink*. Examples of these types of *Source* devices may include pulse oximeters, EEGs, and ECGs.

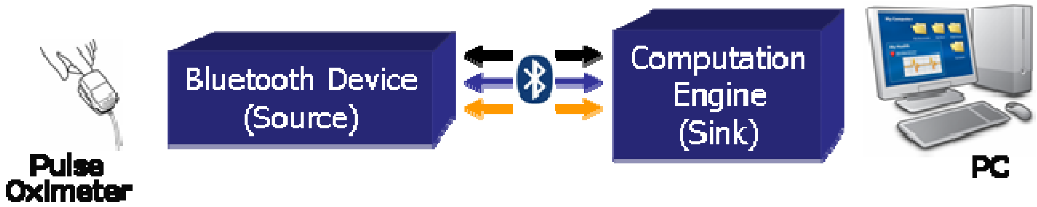

Figure 2.4: Example HDP Use - Streaming Data

Figure 2.5 shows an example scenario in which application data from *Source* devices with both Reliable and Streaming Data Channels is transmitted to a *Sink*.

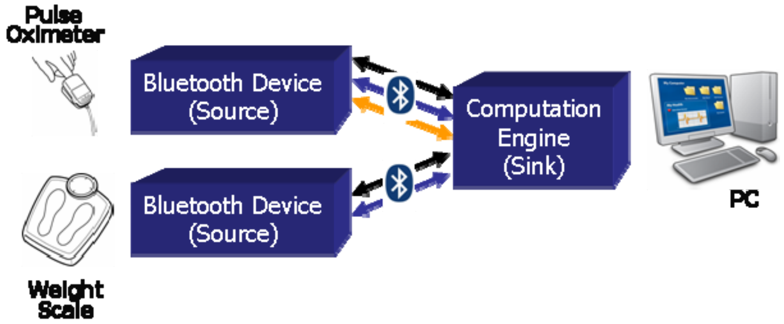
Pulse Oximeter
Bluetooth Device (Source)
Computation Engine (Sink)
PC
Bluetooth Device (Source)
Weight Scale

Figure 2.5: Example HDP Use - Reliable and Streaming Data

Figure 2.6 shows an example scenario in which application data from multiple *Sources* is transmitted to a *Sink,* which facilitates the ability of the application data to be viewed and analyzed by a physician or remote care provider via alternate transports. The shaded portion of this diagram is beyond the scope of this specification.

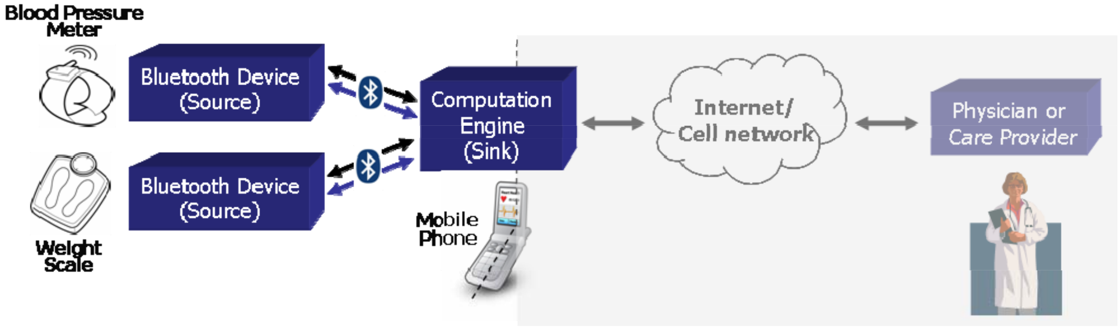
Blood Pressure Meter
Bluetooth Device (Source)
Bluetooth Device (Source)
Weight Scale
Computation Engine (Sink)
Mobile Phone
Internet/ Cell network
Physician or Care Provider

Figure 2.6: Example HDP Use - Sharing of Application Data with Remote Care Provider

Figure 2.7 shows an example scenario in which application data from multiple *Sources* is transmitted to a *Sink* that is also able to take on the role of a *Source* for transmission of logged application data or alarm information to a *Sink* as needed. In this example, a patient Display Console acts as a *Sink* for patient sensors and as a *Source* for a Central Nurse Station PC where all patients are monitored. This Dual Role Device is able to transparently forward the information from the initial *Source* devices to the Computation Engine. The Computation Engine sees the *Source* of the application data as the initial *Source* devices, not the Dual Role Device.

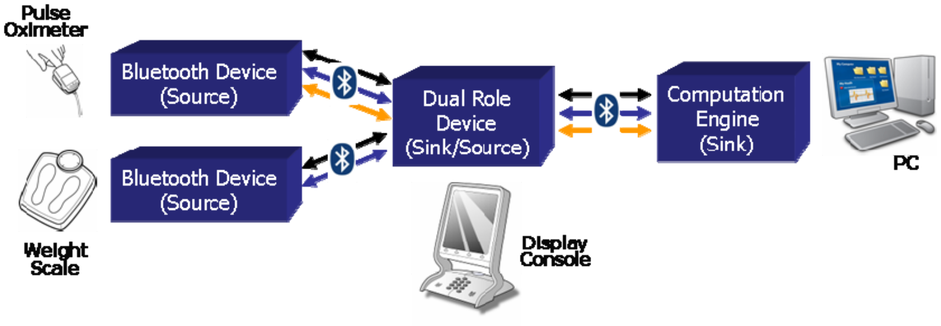
Pulse Oximeter
Bluetooth Device (Source)
Bluetooth Device (Source)
Weight Scale
Dual Role Device (Sink/Source)
Display Console
Computation Engine (Sink)
PC

Figure 2.7: Example HDP Use - Intermediate Source/Sink *Device*

Figure 2.8 shows an example scenario in which application data from a combination *Source* device is transmitted to a *Sink*. This combination device may require multiple Data Channels which may need to be configured differently. In this example, streaming waveform data is sent on a Streaming Data Channel and periodic temperature data is sent on a Reliable Data Channel.

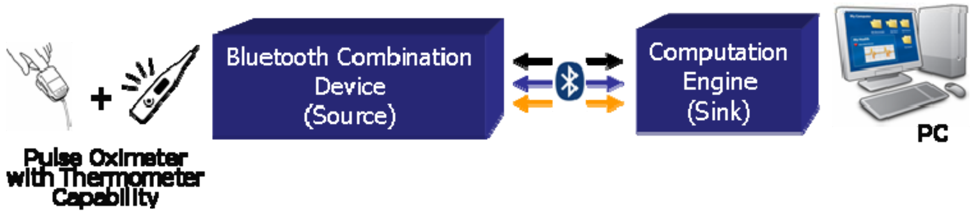
Pulse Oximeter with Thermometer Capability
Bluetooth Combination Device (Source)
Computation Engine (Sink)
PC

Figure 2.8: Example HDP Use - Combination Source Device

Further examples of HDP usage may include:

1. A blood pressure unit (*Source*) is used to measure blood pressure. After measuring the result, the *Source* searches for and discovers the *Sink* device that is to receive its blood pressure data. The *Source* initiates a connection, and the application data is transferred. In this case, the *Source* is the *Initiator*, and the *Sink* is the *Acceptor*.
2. A computation engine (*Sink*) discovers a previously unknown portable medication compliance device (i.e. pill dispenser) which is configured to record activities and does not perform periodic device discovery itself (and will therefore not notice when the *Sink* is available to receive an application data transmission). The *Sink* connects and requests a list of medication events from a specified period of time. In this case, the *Sink* is the *Initiator*, and the *Source* is the *Acceptor*. Although this example also assumes that the *Source* can support queries that specify time ranges, the actual set of acceptable queries is defined in the Data Exchange Specifications.
3. A Heart Rate Monitor device implements a *Source* MDEP to send application data to a cell phone that has implemented a *Sink* MDEP.
4. An Electro-Cardiogram (ECG) device implements a *Source* MDEP and streams application data to a PC that has implemented a *Sink* MDEP.
5. A PC that implements many *Sink* MDEPs can collect application data from several devices that have each implemented a *Source* MDEP.
6. A PC that implements both the *Sink* and the *Source* MDEP can act as a device gateway that receives application data from several *Source*s and retransmits the application data and device status information to another device that has implemented a *Sink* MDEP.
7. A caregiver makes changes to an alarm/alert threshold in a device with a Computation Engine that implements the *Sink* MDEP, and these changes are relayed to a sensor device that implements the *Source* MDEP.
8. A patient-bedside monitor can act as a *Sink* for patient worn sensors and a *Source* for a Central Nurse's Station where all patients are monitored.

3 Protocol Layer

The underlying MCAP layer provides for the establishment of an MCL and whatever component MDLs are required. Each MDL provides a Data Channel connection between two devices, over which application data compliant with the Data Exchange Protocol and Device Data Specializations are transmitted.

MCAP defines that the initial connection of an MDL begins with the creation of an MDL ID, the opening of an L2CAP channel as the Data Channel, and the initialization of the application data connection as defined in the Data Exchange Specifications.

3.1 Generic Access Profile (GAP) Configuration

3.1.1 GAP Modes

The Mode Procedures as defined in GAP [1] describe requirements for both Bluetooth devices involved. The HDP further refines the requirements.

- General-discoverable mode **shall** be supported by all *Sources* and *Sinks*.
- Bondable mode (referred to as Pairable mode prior to 2.1 + EDR) **shall** be supported by all *Sources* and *Sinks*.

Mode Procedure	GAP Status [1]	Support in Source	Support in Sink
General-discoverable mode	O	M	M
Bondable mode	C.1	M	M

Table 3.1: Supported GAP Modes

C.1: If initiation of Bonding is supported (see Section 3.1.3), support for Bondable mode is Mandatory, otherwise Optional.

3.1.2 Security Aspects

HDP requirements for security are described in the following sections.

3.1.2.1 Authentication and Encryption Requirements

- To ensure secure data transfer, all Data Channel and Control Channel connections **shall** always be on authenticated and encrypted links.
- Response to Authentication **shall** be supported by all *Sources* and *Sinks*.
- Initiation of Authentication **shall** be supported by all *Sources* and *Sinks*.

Authentication Procedure	GAP Status [1]	Support in Source	Support in Sink
Response to Authentication request	M	M	M
Initiation of Authentication	C.1	M	M

Table 3.2: Supported Authentication Features

C.1: Mandatory if Security Mode 2, 3, or 4 is supported, otherwise Optional

- Acceptance of Encryption requests **shall** be supported by all *Sources* and *Sinks*.

- Initiation of Encryption **shall** be supported by all *Sources* and *Sinks*.

Encryption Procedure	LMP Status [1]	Support in Source	Support in Sink
Acceptance of Encryption request	M	M	M
Initiation of Encryption	C.1	M	M

Table 3.3: Supported Encryption Features

C.1: Mandatory if Bluetooth version 2.1 + EDR or later is claimed, otherwise Optional.

3.1.2.2 Additional Security Requirements

The Bluetooth 2.1 + EDR specification mandates the use of Secure Simple Pairing. This is very useful for health devices, as it allows an easier and more secure exchange of authentication and encryption credentials. For pairing guidance using version 2.1 + EDR and earlier devices, please refer to the Version 2.1 + EDR Bluetooth Core Specification [1] and the Bluetooth User Interface Flow Diagrams for Bluetooth Secure Simple Pairing Devices [9].

- Devices implementing Bluetooth 2.1 + EDR or later **should** implement man-in-the-middle protection.

Security against sniffing attacks is important in health devices, and the use of PIN lengths of at least 6 digits can improve protection in this respect.

- HDP devices implementing 2.0 + EDR and earlier **should** use PIN lengths of at least 6 digits. Bluetooth 2.1 + EDR and later specifications mandate support for Secure Simple Pairing (SSP) which provides adequate protection against sniffing attacks.

3.1.3 Idle Mode Procedures

The table below shows the support status for Idle Mode Procedures for *Sinks* and *Sources*. For definitions of identifiers used in this table, see Section 1.4.1.

The Idle Mode procedures as defined in GAP [1] describe requirements for both Bluetooth devices involved. The HDP further refines the requirements.

- General inquiry **shall** be supported by all *Sinks* and **should** be supported by *Sources*.
- The Bonding procedure as defined in GAP [1] describes requirements for both Bluetooth devices involved. The HDP further refines the requirements.
- Acceptance of Bonding **shall** be supported by all *Sources* and *Sinks*
- Initiation of Bonding **should** be supported by *Sources* and *Sinks*.

Idle Mode Procedure	GAP Status [1]	Support in Source	Support in Sink
General inquiry	C.1	C.2	M
Acceptance of Bonding request	O	M	M
Initiation of Bonding	O	O	O

Table 3.4: Supported Idle Mode Procedures

C.1: If initiation of Bonding is supported, support for at least one inquiry procedure is Mandatory, otherwise Optional.

C.2: If initiation of Bonding is supported, support for General inquiry procedure is Mandatory, otherwise Optional.

3.1.4 Extended Inquiry Response Requirements

Extended Inquiry Response (EIR) is a feature introduced in Bluetooth Core Specification version 2.1 + EDR or later.

Since General-discoverable mode is specified as optional in GAP, but mandatory in HDP (refer to Section 3.1.1), all *Sources* and *Sinks* implementing Bluetooth Core Specification 2.1 + EDR or later are required to support EIR.

- HDP devices **should** use EIR to facilitate the device selection process (i.e. HDP, HDP Source and HDP Sink) upon general discovery.
- Device Identification tags **should** be included in the EIR. Refer to Section 5.3 for Device ID requirements.

Refer to the Device Identification specification [11] for optional EIR tags.

3.1.5 Class of Device Requirements

The following are requirements related to the Class of Device (CoD) field:

- Determination of whether a device supports HDP, or whether it supports the *Source* and/or *Sink* role(s) of HDP, **shall only** be made using either the ServiceClassID obtainable via SDP or Extended Inquiry Response (EIR). The CoD **shall not** be used for this purpose.
- *Sources* **should** indicate ‘Health’ as the Major Device Class and *Sinks* **may** indicate ‘Health’ as the Major Device Class.
  - If a *Source* or *Sink* indicates ‘Health’ as the Major Device Class, it **shall** indicate an appropriate Health Minor Device Class.

Refer to the Bluetooth Assigned Numbers [3] for more information on CoD fields.

3.1.6 Role Switch

The HDP specification does not place any requirements for role selection.

3.2 MCAP Requirements

This section defines MCAP feature requirements for HDP. Unless otherwise stated in the following sections, requirements are as specified in MCAP [2].

MCAP Feature	MCAP Status [2]	Support in HDP
Support for Standard Op Codes	O	M
Support for Clock Synchronization Protocol	O	O

Table 3.5: MCAP Feature Support

MCAP [2] defines not only the Data Channel establishment procedures, but also a Clock Synchronization Protocol (CSP).

- If clock synchronization is performed over Bluetooth between two devices that support HDP, it **shall** use the CSP defined in MCAP.

If the Clock Synchronization Protocol is supported, refer to the SDP record requirement in Section 5.2.11.

For a more detailed description of the Clock Synchronization Protocol including usage examples, please refer to the HDP Implementation Guidance Whitepaper [12].

3.3 L2CAP Control Channel Requirements

This section defines the L2CAP Control Channel requirements for HDP. Unless otherwise stated in the following sections, L2CAP requirements are as specified in the Core Specification [1].

The Control Channel benefits from useful features of L2CAP [1] including: Protocol/channel multiplexing, Segmentation and Reassembly, flow control, error control retransmissions, Enhanced Retransmission Mode and FCS. The requirements and implementation details of these beneficial features are discussed throughout this section.

L2CAP provides mechanisms to configure channels in such a way as to guarantee reliable delivery of data where “reliable” means that retransmissions are done by L2CAP in an attempt to deliver data without errors or losses and “guaranteed” means that data is to be delivered, up to the point where the entire connection is dropped. In the case where a data connection is dropped, reconnection is described in Section 3.5 of this document.

In order to ensure interoperability between *Sources* and *Sinks*, the following requirement applies:

- Devices that advertise an SDP record with an HDP entry, **shall** accept a connection to the Control Channel Protocol/Service Multiplexer [1] (PSM) that is indicated in that SDP record. Note that this includes all *Sinks.* MCAP mandates that all *Sinks* accept Control Channel connections, for which HDP requires an SDP record that advertises the Control Channel PSM.
- Control Channel PSMs **shall** be Dynamic PSMs [1].
- In order to avoid assumptions regarding the value of Control Channel PSMs, PSM values **shall** be retrieved from the SDP record.

Marking Control Channel packets as non-flushable will prevent the inadvertent flushing of ACL packets. The 2.1 + EDR core specification added a feature called Non-Flushable Packet Boundary Flag for this purpose.

3.3.1 Configuration of L2CAP Control Channel

For the Control Channel, Enhanced Retransmission Mode and FCS (defined in the Core Specification [1]) are used such that any lost or corrupted packets will be detected and retransmitted by the retransmission protocol.

- The Control Channel **shall** use Enhanced Retransmission Mode.
- The Control Channel **shall** have the FCS enabled.
- TxWindow size of 1 **should** be used for the Control Channel because it is sufficient for a request/response protocol.
- The value for MaxTransmit **should** be set to a large enough value to avoid premature disconnects due to transient radio interference but short enough to determine a link is not making progress in a reasonable amount of time.

Note that Enhanced Retransmission Mode provides per-L2CAP-channel flow control, Segmentation and Reassembly (SAR) and per-L2CAP-channel error control and retransmissions.

3.3.2 Maximum Transmission Unit (MTU)

The Core Specification [1] requires that all L2CAP implementations support a minimum MTU size of 48 bytes. Since the largest defined Control Channel message is 16 bytes, an MTU size of 48 bytes will not restrict Control Channel messages.

3.3.3 L2CAP Requirements Summary for Control Channel

The table below provides a summary of supported L2CAP Control Channel requirements for all *Sinks* and *Sources*.

L2CAP Control Channel Configuration Parameter	L2CAP Status [1]	Support in Control Channel
Enhanced Retransmission Mode	C.1	M
Uses FCS for Control Channel	N/A	M

Table 3.6: L2CAP Control Channel Configuration Summary

C.1: At least one of Enhanced Retransmission Mode or Streaming Mode is Mandatory to support.

3.3.4 Quality of Service

Negotiation of Quality of Service for the Control Channel is optional in HDP.

3.4 L2CAP Data Channel Requirements

This section defines L2CAP Data Channel requirements for the HDP. Unless otherwise stated in the following sections, L2CAP requirements are as specified in the Core Specification [1].

The Data Channels benefit from useful features of L2CAP [1] including: Protocol/channel multiplexing, Segmentation and Reassembly, per-L2CAP-channel flow control, per-L2CAP channel error control retransmissions, Enhanced Retransmission

Mode, Streaming Mode and optional FCS. The requirements and implementation details of these beneficial features are discussed throughout this section.

L2CAP provides mechanisms to configure channels in such a way as to guarantee reliable delivery of data or to configure channels to allow them to be flushed for applications relating to streaming or Guaranteed Latency service type where packets can be discarded if they cannot be delivered within a specific time interval. In this context, “reliable” means that retransmissions are done by L2CAP in an attempt to deliver data without errors or losses and “guaranteed” means that data is to be delivered, up to the point where the entire connection is dropped. In the case where a data connection is dropped, reconnection is described in MCAP [2].

- Data Channels **shall** be configured either to be Reliable or Streaming according to the requirements defined in Sections 3.4.1 and 3.4.2. The decision for which Data Channel configuration is used for transport of application data is left to the implementation, however the implementer is encouraged to refer to the appropriate Device Data Specialization specifications for guidance.
- All *Source* and *Sink* implementations **shall** support at least one Reliable Data Channel.

Because packet-loss during the association procedure of IEEE 11073-20601 can cause long delays from the built-in loss-recovery timeouts, a Reliable Data Channel provides important performance advantages.

- The first Data Channel opened on a given MCL **shall** be a Reliable Data Channel.
- Association traffic **shall** be carried on the first Reliable Data Channel that was opened on a given MCL.

> Note: For purposes of this requirement, “operational traffic” refers to packets that are sent with the expectation that both the sender and the receiver were in the IEEE 11073-20601 “Operating” state before the transmission, and will remain in the “Operating” state after the packet has been correctly processed. “Association traffic” refers to all packets that do not fit this definition of “operational traffic.”

- *Sinks* **shall** support Streaming Data Channels.
- *Sources* **may** support Streaming Data Channels.
- Data Channels **shall** support an MTU (following reassembly of PDUs by SAR) large enough to receive the largest packet their specialization (and role as Manager or Agent) may expect them to receive, as defined in the Device Data Specialization specifications.

MCAP allows a Configuration value to be included in MD_CREATE_MDL request and response operations. The values used by HDP are shown in Table 3.7.

Value	Data Channel Configuration
0x00	MCAP Reserved “no preference” value
0x01	Reliable Data Channel
0x02	Streaming Data Channel
0x03 - 0xFF	Reserved

Table 3.7: Data Channel Configuration values for use with MD_CREATE_MDL operations

- When initiating a MD_CREATE_MDL operation, the *Source* **shall** indicate a preference for either a Streaming Data Channel or Reliable Data Channel in the Configuration field of the request packet.
  - If a MD_CREATE_MDL operation succeeds, but the Configuration value returned in the MD_CREATE_MDL response (Response Parameter field of the response packet) does not match the indicated preference, the *Source* **shall** close the MCL.
- When initiating a MD_CREATE_MDL operation, the *Sink* **shall** indicate a Configuration value of “no preference” (0x00).
  - If a Data Channel that uses the Configuration value returned in the MD_CREATE_MDL response cannot be opened by the *Sink*, the *Sink* **shall** either close the MCL or use an Abort operation for the MDL ID it was trying to create.
- When accepting a MD_CREATE_MDL operation with a zero (“no preference”) Configuration value, a *Source* **shall** respond with a preference value indicating either Reliable or Streaming.
  - When accepting a MD_CREATE_MDL operation with a non-zero Configuration value, a *Source* **shall** respond with a Configuration Rejected response.
- When accepting a MD_CREATE_MDL operation with a Configuration value indicating either Reliable or Streaming, a Sink **may** reject the connection (with a Configuration Rejected response) if it is unable to open a Data Channel of that type (e.g. due to resource constraints).
  - When accepting a MD_CREATE_MDL operation with a zero (“no preference”) Configuration value, a *Sink* **shall** respond with a Configuration Rejected response.
- If a *Source* or *Sink* receives an MD_CREATE_MDL request with a non-zero Configuration value that is not included in Table 3.7, the MCL **shall** be closed.

Refer to Section 3.5 for requirements related to reconnection.

In order to ensure interoperability between *Sources* and *Sinks*, the following requirement applies:

- Devices that advertise an SDP record with an HDP entry (note that this includes all *Sinks*), **shall** accept connections to the Data Channel PSM and MDEP IDs that are indicated in that SDP record. Note that this includes all *Sinks.* MCAP mandates that all *Sinks* accept Data Channel connections, for which HDP requires an SDP record that advertises the Data Channel PSM.
- Data Channel PSMs **shall** be Dynamic PSMs [1].
- In order to avoid assumptions regarding the value of Data Channel PSMs, PSM values **shall** be retrieved from the SDP record.

The presence of an active Streaming Data Channel implies that Data Channel packets may need to be flushed at the ACL link. When Streaming and Reliable Data Channels share the same ACL link, flushing can result in temporary loss of a few reliable channel packets, however, Enhanced Retransmission Mode in the Reliable Data Channel will eventually re-transmit the flushed packets and the Data Channel will still maintain reliability.

Marking Reliable Data Channel packets as non-flushable will prevent the inadvertent flushing of ACL packets. The 2.1 + EDR core specification added a feature called Non-Flushable Packet Boundary Flag for this purpose.

3.4.1 Configuration of Reliable Data Channels

For Reliable Data Channels, Enhanced Retransmission Mode (defined in the Core Specification [1]) is used such that any lost or corrupted packets will be detected and retransmitted by the retransmission protocol.

- All Reliable Data Channels **shall** use Enhanced Retransmission Mode.
- All Reliable Data Channels **shall** be capable of sending data using SAR
- To enable *Sources* to have the flexibility to set the FCS if the *Sink* has no preference, Reliable Data Channels on the *Sink* **should** be configured to use the FCS Option with a value of "No FCS". Because the Core Specification [1] specifies that 16-bit FCS is to be used by default, the only way that no FCS can be used is if both sides send the FCS Option with a value of "No FCS".
- The value for MaxTransmit **should** be set to a large enough value to avoid premature disconnects due to transient radio interference but short enough to determine a link is not making progress in a reasonable amount of time.

Note that Enhanced Retransmission Mode provides per-L2CAP-channel flow control, Segmentation and Reassembly (SAR) and per-L2CAP-channel error control and retransmissions.

The table below provides a summary of supported Reliable Data Channel configuration requirements.

L2CAP Data Channel Configuration Parameter	L2CAP Status [1]	Support on Reliable Data Channel(s)
Enhanced Retransmission Mode	C.1	M
Can use FCS Option of “No FCS”	C.2	O
Can send data using SAR in ERTM	C.3	M

Table 3.8: Reliable Data Channel Configuration Summary

C.1: At least one of Enhanced Retransmission Mode or Streaming Mode is Mandatory to support.

C.2: Optional if Enhanced Retransmission Mode or Streaming Mode is claimed, otherwise Excluded.

C.3: Optional if Enhanced Retransmission Mode is claimed, otherwise Excluded.

3.4.2 Configuration of Streaming Data Channels

For Streaming Data Channels, Streaming Mode (defined in the Core Specification [1]) is used such that packets can be discarded if they cannot be delivered within a specific time. Streaming Mode requires the use of a finite flush timeout for the channel.

- All Streaming Data Channels **shall** use Streaming Mode.
- All Streaming Data Channels **shall** be capable of sending data using SAR
- To enable *Sources* to have the flexibility to set the FCS if the *Sink* has no preference, Streaming Data Channels on the *Sink* **should** be configured to use the FCS Option with a value of "No FCS". Because the Core Specification [1] specifies that 16-bit FCS is to be used by default, the only way that no FCS can be used is if both sides send the FCS Option with a value of "No FCS".
- Streaming Data Channels on *Sinks* **should** be configured to have a TxWindow as large as the *Sink* can support to protect against flushed packets and to maximize channel utilization.

Note that Streaming Mode provides Segmentation and Reassembly (SAR).

The table below provides a summary of supported Streaming Data Channel configuration requirements.

L2CAP Data Channel Configuration Parameter	**L2CAP Status [1]**	**Support on Streaming Data Channel(s)**
Streaming Mode	C.1	M
Can use FCS Option of “No FCS”	C.2	O
Can send data using SAR in SM	C.3	M

Table 3.9: Streaming Data Channel Configuration Summary

C.1: At least one of Enhanced Retransmission Mode or Streaming Mode is Mandatory to support.

C.2: Optional if Enhanced Retransmission Mode or Streaming Mode is claimed, otherwise Excluded.

C.3: Optional if Streaming Mode is claimed, otherwise Excluded.

3.4.3 Maximum Transmission Unit (MTU)

The Bluetooth Core Specification has specified the “default” MTU as 672 bytes, however, due to the extreme memory resource constraints faced by some embedded devices, it is expected that some devices are unable to support an MTU size this large. The Core Specification [1] requires that all L2CAP implementations support a minimum MTU size of 48 bytes. Larger MTU sizes for the Data Channel will aid with the transmission of larger amounts of application data. See Section 4.1.3 for Data Channel framing requirements for cases where the MTU is larger than the L2CAP MPS (Max PDU Payload Size).

3.4.4 Quality of Service

Negotiation of Quality of Service for Data Channels is optional in HDP.

3.5 Reconnections

- To allow devices with state memory to save themselves redundant configuration steps, a device **may** send a reconnect request to an MDL ID that has already been established.
- When using a MD_RECONNECT_MDL operation, the last negotiated Data Channel L2CAP configuration **shall** be used for the reestablished channels. The configuration parameters referred to here include (but are not limited to) MTU and Operating Mode (Streaming Mode or Enhanced Retransmission Mode).
  - In case the last negotiated L2CAP channel configuration is not achieved upon reconnect, the L2CAP Data Channel **shall** be disconnected.

MCAP provides a reconnect feature for requesting a connection to a pre-existing context. If the device receiving the reconnect request has retained enough state to allow the MDL to be reopened, the reconnect is accepted. If the device receiving the request has not retained the required state information (possibly due to a lack of stable storage, intervening firmware upgrades, etc.) the reconnect is rejected.

The MCAP Supported Procedures attribute defined in Section 5.2.11 includes bits called “Supports Reconnect Initiation” and “Supports Reconnect Acceptance”. If these bits are set, this indicates that the respective device is capable of maintaining state information. This information can be used by the reconnecting device to improve efficiency of reconnect operations.

4 Data-Layer Requirements

4.1 Overview

As specified in MCAP [2], a Control Channel is used to transmit control requests between a *Sink* and a *Source* device. One or more Data Channels are used to transmit and receive device-specific application data to and from a *Source* device. The contents of the Data Channel, beyond the framing bytes defined in Section 4.1.3 below, are specified by the Data Exchange Specifications. An implementation of HDP uses MCAP to facilitate the establishment of these Data Channels (although HDP defines their required configuration), providing a transport mechanism to allow the data-layers to exchange application data. The format of that application data is specified in the appropriate Device Data Specialization specification(s) and the exchange protocol is specified in the Data Exchange Protocol [7] specifications.

When a Data Channel is opened, the format of its contents is defined completely by the Data Exchange Specifications. During channel establishment, per-channel L2CAP configuration options (Error Control, etc.) are established for each MDL. HDP makes use of the L2CAP facility that a single Protocol/Service Multiplexer [1] (PSM) (end point) can be used to initiate a plurality of simultaneous connections, each with independent attributes.

4.1.1 Data Exchange Specification Requirements

The Data Exchange Specification to be used with the HDP is defined by the Data Exchange Specification attribute in the SDP record. Refer to Section 5.2.10 for information regarding this attribute.

4.1.2 Device Data Specialization Requirements

The Device Data Specialization to be used with the HDP is defined by the MDEP Data Type attribute in the SDP record. Refer to Section 5.2.9.2 for information regarding this attribute.

4.1.3 Data Channel Framing

As the introduction to this document explains, HDP is designed to work with the ISO/IEEE 11073-20601 Data Exchange Protocol [7], which is packet-based. Although L2CAP is also packet-based, physical restrictions sometimes limit the MTU options for a particular implementation. Because the 11073-20601 protocol requires packets to have the same boundaries with which they were created, some requirements are placed on the Data Channel configuration.

In most cases, Device Data Specialization specifications define the maximum packet size for each device specialization. These are usually asymmetric (typically with larger maximum sizes for packets from *Source*-to-*Sink* than in the other direction). Implementers are encouraged to refer to these specifications for guidance.

- Some data-layer packets may be larger than the L2CAP MPS (Max PDU Payload Size) of the Data Channel. L2CAP Segmentation and Reassembly (SAR) **shall** be used to maintain packet framing in this situation.
- The Data Channel **shall** support the transmission and receipt of packets up to the size required by the transmit and receive packet limits of the appropriate Device Data Specialization such that they are relayed to the data-layer with their original packet boundaries (un-fragmented).
- If multiple Device Data Specializations are supported, all of the various maximum packet sizes in those specializations **shall** be supported.
- Partially-received packets **shall not** be reported to the data-layer.

5 Service Discovery

Support for SDP Server is a prerequisite for devices to be able to advertise their SDP Record (and PSMs) thus being able to accept Control Channel or Data Channel connections and reconnections.

Support for SDP Client is a prerequisite for devices to be able to retrieve the SDP Record (and PSMs) from a remote device thus being able to initiate Control Channel or Data Channel connections and reconnections.

Refer to Sections 3.3 and 3.4 for requirements related to advertisement of an SDP record.

Refer to Section 5.3 for Device ID requirements.

5.1 Service Record for HDP Device

In order to establish a flexible scheme, SDP includes the Protocol/Service Multiplexer (PSM - as defined in the L2CAP specification [1]) in the protocol parameters for the L2CAP layer.

Although two PSMs are used (one for the Control Channel, and the other for Data Channels), only one SDP record is required. The Control Channel PSM is identified in the Protocol Descriptor List, and the Data Channel PSM is found in the first (and only) element of the Additional Protocol Descriptors list.

- A second SDP record is considered a valid configuration and **may** be used for cases where a device runs more than one HDP implementation (i.e. with own unique Control Channel and Data Channel PSMs).

The codes assigned to the mnemonics used in the “Value” column as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column) can be found in the Bluetooth Assigned Numbers [3]. Please refer to the relevant section of the Bluetooth Assigned Numbers [3] for the most updated SDP information.

- When present, an SDP Record for an HDP implementation **shall** support all mandatory attributes in Table 5.1 and use the correct reserved values as specified in the Bluetooth Assigned Numbers [3]. Those attributes not described in this chapter are defined in the SDP section of the Core Specification [1].

Item	Definition	Type	Value	AttrID	Status
Service Class ID List		Sequence		0x0001*	M
Service Class #0		UUID	Variable		M
Protocol Descriptor List		Sequence		0x0004*	M
Protocol #0		Sequence			M
Protocol #0 Identifier	L2CAP	UUID	L2CAP		M
Parameter #0 for Protocol #0	PSM	UINT 16	Control Channel PSM		M
Protocol #1		Sequence			M
Protocol #1 Identifier	MCAP	UUID	MCAP Control Channel		M
Parameter #0 for Protocol #1	Version	UINT 16	0x0100		M
Bluetooth Profile Descriptor List		Sequence		0x0009*	M
Profile #0		Sequence			M
Profile #0 Identifier	HDP	UUID	HDP		M
Parameter #0 for Profile #0	Version	UINT 16	0x0100		M
Additional Protocol Descriptor Lists		Sequence		0x000D*	M
Protocol Descriptor List #0		Sequence			M
Protocol #0		Sequence			M
Protocol #0 Identifier	L2CAP	UUID	L2CAP		M
Parameter #0 for Protocol #0	PSM	UINT 16	Data Channel PSM		M
Protocol #1		Sequence			M
Protocol #1 Identifier	MCAP	UUID	MCAP Data Channel		M
Service Name	Displayable Text Name	String	Configurable	0x0100 + 0x0000*	O
Service Description	Displayable Text Name	String	Configurable	0x0100 + 0x0001*	O
Provider Name	Displayable Text Name	String	Configurable	0x0100 + 0x0002*	O
Supported Features	Supported formats	Sequence		0x0200	M
Supported Feature #0	MDEP List	Sequence			M
MDEP ID	Variable	UINT 8			M
MDEP Data Type	Sec. 5.2.9.2	UINT 16			M
MDEP Role	Table 5.3	UINT 8			M
MDEP Description	Displayable Text Name	String			O
Data Exchange Specification	Table 5.4	UINT 8	0x01	0x0301	M
MCAP Supported Procedures	Table 5.5	UINT 8		0x0302	M

Table 5.1: SDP Record

* For informative purposes only. For definitive Attribute ID values, refer to the Bluetooth Assigned Numbers [3].

5.2 Service Record Attribute Details

Most elements of a HDP service discovery record are simply used as described in the SDP specification. Some that require definition, or can benefit from further detail are discussed in this section.

5.2.1 UUIDs

Refer to the Bluetooth Assigned Numbers [3] for these values.

5.2.2 Service Class ID List

The Service Class ID list contains an entry identifying the profile and role supported by the implementation.

- The HDP *Source* Service Class ID **shall** be present if and only if one or more Service Class records indicate support for the *Source* role.
- The HDP *Sink* Service Class ID **shall** be present if and only if one or more Service Class records indicate support for the *Sink* role.

Acceptable entries in the Service Class ID List attribute of an HDP service are HDP *Source* Service Class and HDP *Sink* Service Class.

- The Service Class ID List **shall** contain one or two entries
- The Service Class ID List entries **shall** be chosen from the set of “HDP *Source* Service Class” and “HDP *Sink* Service Class,” as defined in the Bluetooth Assigned Numbers [3].

5.2.3 Protocol Descriptor List

The Protocol Descriptor List entry is as defined in the Core Specification [1] with a version number associated with the MCAP identifier. This identifies the version of MCAP being used (i.e. for both the Control Channel and the Data Channel).

- The MCAP version number **shall** be identified in this entry.

5.2.4 Bluetooth Profile Descriptor List

As stated in the Bluetooth Core specification [1], the BluetoothProfileDescriptorList attribute consists of a data element sequence in which each element is a profile descriptor that contains information about a Bluetooth profile to which the service represented by this service record conforms. Each profile descriptor is a data element sequence whose first element is the UUID assigned to the profile and whose second element is a 16-bit profile version number.

As specified in the Bluetooth Core specification, each version of a profile is assigned a 16-bit unsigned integer profile version number, which consists of two 8-bit fields. The higher-order 8 bits contain the major version number field and the lower-order 8 bits contain the minor version number field. When upward compatible changes are made to the profile, the minor version number will be incremented. If incompatible changes are made to the profile, the major version number will be incremented.

- This version of the HDP **shall** indicate version number 0x0100 (representing version 01.00), and the profile UUID as assigned in the Bluetooth Assigned Numbers [3].

5.2.5 Additional Protocol Descriptor Lists

The Additional Protocol Descriptor List is defined in the SDP specification (version 2.1 + EDR or later)[2], and contains a sequence of sequences. The contained sequences (each element of the data element sequence that constitutes the value associated with this attribute) are formatted according to the specification of the Protocol Descriptor attribute (Attribute ID 0x0004), constituting a list of protocol UUIDs and associated parameters ordered from most general to most specific.

In the case of HDP, the Additional Protocol Descriptor List attribute is mandatory. As consistent with the definition in the Core Specification, this attribute contains a sequence having one entry.

- The Additional Protocol Descriptor List attribute **shall** contain only one entry, and that entry **shall** be a Protocol Descriptor for the Data Channel L2CAP connection, as shown in Table 5.1.
- The MCAP portion of this entry **shall not** contain a version number (refer to the Protocol Descriptor List entry in Section 5.2.3).

5.2.6 Service Name

The ServiceName attribute is a string containing the name of the service represented by a service record. It should be brief and suitable for display such as with an Icon representing the service. Because more than one HDP implementation may exist on a single machine, the Service Name attribute allows distinct HDP services to be differentiated.

- An implementation of HDP that allows multiple HDP services to share a Bluetooth address **shall** ensure that the SDP records for each of these services contain Service Name entries, and that these entries are distinct from one another.

5.2.7 Service Description

The ServiceDescription attribute is a string containing a brief description of the service.

- Per the Bluetooth Core Specification [1], it **should** be less than 200 characters in length.

[2] For implementations prior to 2.1 + EDR, implementers may want to refer to Errata numbers 208 and 663 for clarification on use of the Additional Protocol Descriptor List.

5.2.8 Provider Name

The ProviderName attribute is a string containing the name of the person or organization providing the service.

5.2.9 Supported Features (MDEP List)

This is a sequence for which each element is a sequence that describes a single application data end-point on the device. The Supported Features attribute (MDEP List) provides an indication of the data types that an MDEP supports.

- Note that, while each MDEP **shall** have at least one entry in this list, any particular MDEP **may** have more than one entry. For example, if a single MDEP supports both blood pressure and heart rate, then that **shall** be indicated with two entries in the MDEP list.

Symbolically, a Supported Features attribute is a sequence of one or more MDEP definitions, where each description is itself a sequence of three or more elements. The first three elements of a MDEP definition are, respectively, a UINT 8 that identifies the local MDEP ID for the particular MDEP being described, a UINT 16 that identifies the general data type (as defined in the Bluetooth Assigned Numbers [3]), and a UINT 8 that identifies the role (*Source* or *Sink*). If a fourth element is present, it is a String that describes the application data supported by this MDEP in human readable form using the base language of this device (the language used for attribute 0x0100).

- The Supported Features attribute **shall** contain at least one entry.

The elements of each entry are (in order) MDEP ID, MDEP Data Type, MDEP Role, and (optionally) MDEP Description.

- If an implementation changes the contents of this (or any other) attribute, it is important to indicate this change by changing the ServiceRecordState attribute as well. This attribute **should** be used by devices to prevent stale caches of SDP attributes.

5.2.9.1 MDEP ID

An MDEP ID is a coarse indicator that allows an HDP implementation to multiplex communication to various data-layer end points. This is useful for software reasons (e.g. independent coding of end point implementations for 11073-20601 manager and agent on a device that supports both may use logical ports to avoid confusing one end point with traffic intended for the other) or for hardware reasons (e.g. multiple hardware modules have their own data-layer end points but share a single Bluetooth HDP hardware module for communication).

- MDEP ID is an 8-bit value that **shall** be selected to identify an MDEP (data-layer end point) of this HDP implementation.
- Multiple data types **may** be supported by a single data-layer end point. In these cases, one HDP Supported Features entry (using the same MDEP ID) **shall** be used to identify each distinct data type.

Note that MCAP [2] specifies that only one Device Role per MDEP ID is allowed.

The table below shows MDEP IDs for use with HDP.

MDEP ID	Description
0x00	Reserved For Echo Test Function (refer to 5.2.9.1.1)
0x01 - 0x7F	Available for use
0x80 - 0xFF	Reserved by MCAP

Table 5.2: MDEP ID Summary

MDEP ID 0x00 is reserved for an Echo Test Function. This function is defined in the following section.

5.2.9.1.1 Echo Test Function (MDEP ID 0x00)

MDEP ID 0x00 is reserved for debugging and interoperability testing, and is implemented as an Echo Test Function. The general use of this function is that it allows the *Initiator* to send any L2CAP data packet it wants, and the *Acceptor* repeats it back.

- Acceptance of the Echo Test Function **shall** be supported by all *Sources* and *Sinks*.

The Echo Test Function Procedure is defined as follows:

- When an MDL is established for an MDEP ID 0x00, the first communication **shall** be a single data packet (which may be spread across multiple L2CAP packets) and **shall** be sent by the *Initiator* of that MDL connection.
- The *Acceptor* **shall** respond by sending an exact copy of that single data packet.
- Once it has received the response, the *Initiator* **may** close the Data Channel and **may** also close the Control Channel.
- If either device detects an unexpected behavior (more than one data packet is transmitted by the *Initiator*, an incorrect copy is received from the *Acceptor*, etc.), the device detecting that error condition **shall** disconnect the Control Channel (i.e., close the MCL).

5.2.9.2 MDEP Data Type (Device Data Specialization code)

This attribute is a 16-bit value, with the value taken from the Bluetooth Assigned Numbers [3] to identify the Device Data Specialization code used. Note that these values are also referenced in the ISO/IEEE 11073-20601 [7] specification as MDC_DEV_SPEC_PROFILE_* in the Nomenclature Codes Annex. Only those MDEP Data Type attributes listed in the Bluetooth Assigned Numbers [3] are to be used with the HDP to ensure interoperability.

- The value(s) assigned to the MDEP Data Type attribute **shall** be valid according to the Bluetooth Assigned Numbers [3].
- At least one valid Device Data Specialization **shall** be supported.
- All Device Data Specializations claimed in the SDP record, **shall** be supported.

- Device Data Specializations **shall** be defined within the IEEE 11073 Personal Health Data family of standards.
- Device Data Specializations **shall** be compatible with the Data Exchange Protocol identified in the Data Exchange Specification attribute (refer to 5.2.10) to ensure interoperability.
- HDP implementations **shall** have one MDEP Data Type entry for each supported device specialization. (Devices with more than one MDEP Data Type entry are referred to as multi-function devices.)
- The Device Data Specializations identified by this attribute **shall** be implemented in accordance with the external specification which they represent including support for all mandatory features.

5.2.9.3 MDEP Role

- MDEP Role **shall** indicate whether this MDEP is a *Source* of the identified data type, or a *Sink*.
- At least one of the *Source*/*Sink* roles **shall** be supported.

The MDEP Roles definition is shown in the table below:

MDEP Role value	Definition
0x00	Source
0x01	Sink
0x02 - 0xFF	Reserved

Table 5.3: MDEP Role Summary

5.2.9.4 MDEP Description

- MDEP Description is optional. If included, it **shall** give a friendly description of the end-point in the primary language supported by the service record.

5.2.10 Data Exchange Specification

This attribute is a one-byte reference, with the value taken from the Bluetooth Assigned Numbers [3] to identify the Data Exchange Protocol used (e.g. ISO/IEEE 11073-20601 specification).

Valid values for the Data Exchange Specification attribute are maintained in the Bluetooth Assigned Numbers and include the following:

Data Exchange Specification	Document Number	Document Name
0x00	N/A	Reserved
0x01	ISO/IEEE 11073-20601	Health informatics - Personal health device communication - Application profile - Optimized exchange protocol
0x02 - 0xFF	N/A	Reserved

Table 5.4: Data Exchange Specification Summary

- The value assigned to the Data Exchange Specification attribute **shall** be valid according to the Bluetooth Assigned Numbers [3].
- The Data Exchange Specification identified by this attribute **shall** be implemented in accordance with the external specification which it represents.
- *Sources* and *Sinks* **shall** implement all mandatory features of the indicated Data Exchange Specification.
- If a device has more than one instance of a Data Exchange Specification, each instance **shall** be represented as a separate HDP instance each with its own HDP SDP record. As defined previously, multiple Device Data Specializations **may** be included in each HDP SDP record.

The following clarifies expectations between the IEEE 11073-20601 association process (or other Data Exchange Protocol association process) and Bluetooth SDP service and endpoint definitions to ensure alignment. These are to be taken into account when designing the interface between a Data Exchange Protocol and the HDP:

- For MDEP IDs that were used to connect Data Channels prior to the 11073-20601 association process, the association process **shall** allow association to the Device Data Specializations represented by those MDEP IDs. Additional Device Data Specializations **can** be optionally included.
- Each additional MDL that is added to an MCL after an 11073-20601 association process has taken place **shall** be treated as already associated.

5.2.11 MCAP Supported Procedures

This attribute is a one byte bit-mask that indicates the MCAP procedures that are supported by this HDP service.

MCAP Supported Procedures value	Definition
0x00	Reserved
0x01	Reserved
0x02	Supports Reconnect Initiation [3]
0x04	Supports Reconnect Acceptance [4]
0x08	Supports Clock Synchronization Protocol (includes support for at least Sync-Slave Role)
0x10	Supports Sync-Master Role
0x20	Reserved
0x40	Reserved
0x80	Reserved

Table 5.5: MCAP Supported Procedures Summary

[3] If this flag is set, this indicates that this initiating device is capable of maintaining state information.

[4] If this flag is set, this indicates that this accepting device is capable of maintaining state information.

- Multiple MCAP Supported Procedures **shall** be indicated by combining the corresponding values with a logical OR operation. Refer to MCAP [2] and Section 3.2 of this document (MCAP Requirements) for further information regarding the MCAP protocol.
- If a device has an SDP record with an HDP entry, it **shall** indicate the MCAP Supported Procedures as defined in Table 5.5.

5.3 Device ID Requirements

- Devices that advertise an SDP record with an HDP entry (note that this includes all *Sinks*), **shall** implement the Device Identification Profile [11].

6 References

References below are normative unless otherwise specified.

[1] Bluetooth Core Specification 2.0 + EDR or 2.1 + EDR with Volume 3, Part A of Core Specification Addendum 1 or later versions of the Bluetooth Core Specification

[2] Multi-Channel Adaptation Protocol (MCAP)

[3] Bluetooth SIG member web side, Bluetooth Assigned Numbers, http://www.bluetooth.org

[4] Bluetooth Advanced Audio Distribution Profile (A2DP) (Informative)

[5] Bluetooth Basic Imaging Profile (BIP) (Informative)

[6] IEEE Standards Style Manual, http://standards.ieee.org/guides/style/2000Style.pdf

[7] IEEE Std 11073-20601 ™- 2008 Health Informatics - Personal Health Device Communication - Application Profile - Optimized Exchange Protocol - version 1.0 or later

[8] Bluetooth Qualification Reference Document (PRD), version 2.0 or later (Informative)

[9] Bluetooth User Interface Flow Diagrams for Bluetooth Secure Simple Pairing Devices (Informative)

[10] 2.1 + EDR, Volume 0, Part B, Compliance Requirements (Informative)

[11] Bluetooth Device Identification Specification, version 1.3 or later

[12] HDP Implementation Guidance Whitepaper (Informative)

7 List of Figures

8 List of Tables

9 Appendix A: Acronyms and Abbreviations

Abbreviation or Acronym	Meaning
API	Application Program Interface
CoD	Class of Device/Service
CSP	Clock Synchronization Protocol
ECG	Electrocardiogram
EEG	Electroencephalogram
EIR	Extended Inquiry Response
GAP	Generic Access Profile
HDP	Health Device Profile
L2CAP	Logical Link Control and Adaptation Protocol
MCAP	Multi-Channel Adaptation Protocol
MCL	MCAP Communications Link
MDEP	MCAP Device End Point
MDL	MCAP Device Link
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PIN	Personal Identification Number
PSM	Protocol/Service Multiplexer
SAR	Segmentation and Reassembly
SDP	Service Discovery Protocol
UUID	Universally Unique Identifier

10 Appendix B (informative): SDP Record Example

The following sections show SDP Record examples

10.1 MDEP Supported Features Example

Following is an informative example of the MDEP Supported Features fields for a health sensor capable of providing body temperature and blood pressure measurements.

Description	Value	Type
HDP Supported Features Attrib ID 0x200	0x09 0x02 0x00	UINT 16
Data Element Sequence	0x35 0x12	Data elements sequence 18 bytes
First endpoint	0x35 0x07	Data Element Sequence 7 bytes
MDEP ID 0x15	0x08 0x15	UINT 8
MDEP Data Type Body Temperature (0x1008)	0x09 0x10 0x08	UINT 16
MDEP Role 0 (*Source)*	0x08 0x00	UINT 8
Second endpoint	0x35 0x07	Data Element Sequence 7 bytes
MDEP ID 0x15	0x08 0x15	UINT 8
MDEP Data Type Blood Pressure (0x1007)	0x09 0x10 0x07	UINT 16
MDEP Role 0 (*Source*)	0x08 0x00	UINT 8

Table 10.1: MDEP Supported Features Example

10.2 Sensor Service Record Example

This is an informative example of a relay device that reads temperature and blood glucose measurements, displays them locally, and makes these available for transfer (relay) to other reader devices. It has implemented an IEEE 11073-20601 Manager (on MDEP 1) and an IEEE 11073-20601 Agent (on MDEP 2). The services (temperature *Sink* and glucose *Sink*) on MDEP 1 are named (using the optional “MDEP Description” parameter, while the services on MDEP 2 (*Sources* for both data specializations) do not use names for the individual services. The “Service Class ID List” identifies this service as containing both *Source* and *Sink* components, allowing a suitable icon to be chosen on remote devices that may choose to use this service.

Values in YELLOW are general-purpose values defined in the Bluetooth Assigned Numbers [3].

Values in BLUE are defined for the HDP specification only, and their definitive values are also to be found in the Bluetooth Assigned Numbers [3].

Values in GREEN are MDEP Data Types (Device Data Specializations) and Data Exchange Specifications that are defined in the Bluetooth Assigned Numbers [3].

00	**0x36, 0x01, 0x1C**	**sequence, length 284**
03	0x09, 0x00, 0x00	2-byte unsigned integer, 0x0000 (**Attribute 0x0000 = Service Record Handle**)
06	0x0A, 0x04, 0x03, 0x02, 0x01	4-byte integer, the handle for this service
0B	0x09, 0x00, 0x01	2-byte unsigned integer, 0x0001 (**Attribute 0x0001 = Service Class ID List**)
0E	0x35, 0x06	sequence, length 6
10	0x19, 0x14, 0x01	2-byte UUID, 0x1401 (value for HDP Source Service Class)
13	0x19, 0x14, 0x02	2-byte UUID, 0x1402 (value for HDP Sink Service Class)
16	0x09, 0x00, 0x04	2-byte unsigned integer, 0x0004 (**Attribute 0x0004 = Protocol Descriptor List**)
19	0x35, 0x10	sequence, length 16
1B	0x35, 0x06	sequence, length 6 (First Protocol)
1D	0x19, 0x01, 0x00	2-byte UUID, 0x0100 (Protocol ID: L2CAP)
20	0x09, 0x10, 0x01	2-byte unsigned integer (Protocol Parameter 0 [PSM] = 0x1001)
23	0x35, 0x06	sequence, length 6 (Second Protocol)
25	0x19, 0x00, 0x1E	2-byte UUID, 0x001E (Protocol ID: MCAP Control Channel)
28	0x09, 0x01, 0x00	2-byte unsigned integer (Protocol Parameter 0 [version] = 0x0100)
2B	0x09, 0x00, 0x09	2-byte unsigned integer, 0x0009 (**Attribute 0x0009 = Profile Descriptor List**)
2E	0x35, 0x08	sequence, length 8
30	0x35, 0x06	sequence, length 6 (First Profile)
32	0x19, 0x14, 0x00	2-byte UUID, 0x1400 (Profile ID: HDP)
35	0x09, 0x01, 0x00	2-byte unsigned integer (Profile Parameter 0 [version] = 0x0100)
38	0x09, 0x00, 0x0D	2-byte unsigned integer, 0x000D (**Attribute 0x000D = Additional Protocol Descriptor List**)
3B	0x35, 0x0F	sequence, length 15
3D	0x35, 0x0D	sequence, length 13 (First Additional List)
3F	0x35, 0x06	sequence, length 6 (First Protocol)
41	0x19, 0x01, 0x00	2-byte UUID, 0x0100 (Protocol ID: L2CAP)
44	0x09, 0x10, 0x03	2-byte unsigned integer (Protocol Parameter 0 [PSM] = 0x1003)
47	0x35, 0x03	sequence, length 3 (Second Protocol)
49	0x19, 0x00, 0x1F	2-byte UUID, 0x001F (Protocol ID: MCAP Data Channel)
4C	0x09, 0x01, 0x00	2-byte unsigned integer 0x0100 (**Attribute 0x0100 = Service Name**)
4F	0x25, 0x1D	Text String, length 29 "Acme Health Display and Relay" (Service Name)
51	0x41, 0x63, 0x6D, 0x65, 0x20, 0x48, 0x65, 0x61,	
59	0x6C, 0x74, 0x68, 0x20, 0x44, 0x69, 0x73, 0x70,	
61	0x6C, 0x61, 0x79, 0x20, 0x61, 0x6E, 0x64, 0x20,	
69	0x52, 0x65, 0x6C, 0x61, 0x79	
6E	0x09, 0x01, 0x01	2-byte unsigned integer 0x0101 (**Attribute 0x0101 = Service Description**)
71	0x25, 0x27	Text String, length 39 "Collect, display, and relay health data" (Service Description)
73	0x43, 0x6F, 0x6C, 0x6C, 0x65, 0x63, 0x74, 0x2C,	
7B	0x20, 0x64, 0x69, 0x73, 0x70, 0x6C, 0x61, 0x79,	
83	0x2C, 0x20, 0x61, 0x6E, 0x64, 0x20, 0x72, 0x65,	
8B	0x6C, 0x61, 0x79, 0x20, 0x68, 0x65, 0x61, 0x6C,	

93	0x74, 0x68, 0x20, 0x64, 0x61, 0x74, 0x61	
9A	0x09, 0x01, 0x02	2-byte unsigned integer 0x0102 (**Attribute 0x0102 = Provider Name**)
9D	0x25, 0x09	Text String, length 9 "Acme Inc." (Service Description)
9F	0x41, 0x63, 0x6D, 0x65, 0x20, 0x49, 0x6E, 0x63, 0x2E	
A8	0x09, 0x02, 0x00	2-byte unsigned integer 0x0200 (**Attribute 0x0200 = HDP Supported Features**)
AB	0x35, 0x68	sequence, length 104
AD	0x35, 0x2B	sequence, length 43 (First Supported Feature)
AF	0x08, 0x01	1-byte unsigned integer, 0x01 (MDEP = 0x01)
B1	0x09, 0x10, 0x08	2-byte unsigned integer 0x1008 (Specialization 4104 = Body Temperature)
B4	0x08, 0x01	1-byte unsigned integer, 0x01 (Role = Sink)
B6	0x25, 0x22	Text String, length 34 "Acme Thermometer Display and Relay" (Description)
B8	0x41, 0x63, 0x6D, 0x65, 0x20, 0x54, 0x68, 0x65,	
C0	0x72, 0x6D, 0x6F, 0x6D, 0x65, 0x74, 0x65, 0x72,	
C8	0x20, 0x44, 0x69, 0x73, 0x70, 0x6C, 0x61, 0x79,	
D0	0x20, 0x61, 0x6E, 0x64, 0x20, 0x52, 0x65, 0x6C,	
D8	0x61, 0x79	
DA	0x35, 0x27	sequence, length 39 (Second Supported Feature)
DC	0x08, 0x01	1-byte unsigned integer, 0x01 (MDEP = 0x01)
DE	0x09, 0x10, 0x11	2-byte unsigned integer 0x1011 (Specialization 4113 = Glucose Meter)
E1	0x08, 0x01	1-byte unsigned integer, 0x01 (Role = Sink)
E3	0x25, 0x1E	Text String, length 30 "Acme Glucose Display and Relay" (Description)
E5	0x41, 0x63, 0x6D, 0x65, 0x20, 0x47, 0x6C, 0x75,	
ED	0x63, 0x6F, 0x73, 0x65, 0x20, 0x44, 0x69, 0x73,	
F5	0x70, 0x6C, 0x61, 0x79, 0x20, 0x61, 0x6E, 0x64,	
FD	0x20, 0x52, 0x65, 0x6C, 0x61, 0x79	
103	0x35, 0x07	sequence, length 7 (Third Supported Feature)
105	0x08, 0x02	1-byte unsigned integer, 0x02 (MDEP = 0x02)
107	0x09, 0x10, 0x08	2-byte unsigned integer 0x1008 (Specialization 4104 = Body Temperature)
10A	0x08, 0x00	1-byte unsigned integer, 0x00 (Role = Source)
10C	0x35, 0x07	sequence, length 7 (Fourth Supported Feature)
10E	0x08, 0x02	1-byte unsigned integer, 0x02 (MDEP = 0x02)
110	0x09, 0x10, 0x11	2-byte unsigned integer 0x1011 (Specialization 4113 = Glucose Meter)
113	0x08, 0x00	1-byte unsigned integer, 0x00 (Role = Source)
115	0x09, 0x03, 0x01	2-byte unsigned integer, 0x0301 (**Attribute 0x0301 = Data Exchange Specification**)
118	0x08, 0x01	1-byte integer, 0x01 (IEEE 11073-20601)
11A	0x09, 0x03, 0x02	2-byte unsigned integer, 0x0302 (**Attribute 0x0302 = MCAP Supported Procedures**)
11D	0x08, 0x00	1-byte integer, 0x00 (No Reconnect and No Clock Synchronization is Supported)